Amendments to the Claims:

Please amend the claims as follows (it being understood that the following listing of claims will replace all prior versions and listings of claims in the application):

Listing of Claims:

1. (currently amended). A processor verification test apparatus that uses a golden model to generate a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

a user preference queue that comprises queue entries wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

a plurality of resource-related data structures, wherein each said resource-related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

an instruction packer coupled to said user preference queue and said resource-related data structures, said instruction packer creates a group of N instructions valid for parallel execution by the golden model and the processor system under test, where N equals 1 or more; and

an instruction generator and simulator that generates and simulates instructions that correspond to said group of N instructions created by said instruction packer, evaluates the updated architectural state of the golden model, and updates said resource-related data structures.

2. (original). The apparatus of claim 1, wherein said group of N instructions valid for parallel execution further comprises N instructions that do not utilize common system resources other than source registers of the golden model or the processor system under test.

3. (original). The apparatus of claim 1, wherein said instruction packer creates said group of N instructions valid for parallel execution by selecting instructions from an instruction tree in said user preference queue, wherein said instruction packer iteratively creates a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information indicated by said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test.

4. (original). The apparatus of claim 3, wherein said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, and wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction.

5. (original). The apparatus of claim 1, wherein said instruction packer creates said group of N instructions valid for parallel execution by selecting instructions in one of the following ways: by selecting each instruction in the order that said instruction appears in an ordered instruction list in said user preference queue, or by selecting a "no operation" instruction where the next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules.

6. (currently amended). A processor verification test system that uses a golden model to generate a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

a user preference queue that comprises queue entries wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

a plurality of resource-related data structures, wherein each said resource-related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

an instruction packer coupled to said user preference queue and said resource-related data structures, said instruction packer creates a group of N instructions valid for parallel execution by the golden model and the processor system under test, where N equals 1 or more; and

an instruction generator and simulator that generates and simulates instructions that correspond said group of N instructions created by said instruction packer, evaluates the updated architectural state of the golden model, and updates said resource-related data structures.

7. (original). The system of claim 6, wherein said group of N instructions valid for parallel execution further comprises N instructions that do not utilize common system resources other than source registers of the golden model or the processor system under test.

8. (original). The system of claim 6, wherein said instruction packer creates said group of N instructions valid for parallel execution by selecting instructions from an instruction tree in said user preference queue, wherein said instruction packer iteratively creates a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information indicated by said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test.

9. (original). The system of claim 8, wherein said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers that are utilized by said first instruction.

10. (original). The system of claim 6, wherein said instruction packer creates said group of N instructions valid for parallel execution by selecting instructions in one of the following ways: by selecting each instruction in the order that said instruction appears in an ordered instruction list in said user preference queue, or by selecting a "no operation" instruction where the next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules.

11. (currently amended). A method that makes a processor verification test apparatus that uses a golden model to create a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

providing a user preference queue that comprises queue entries wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

providing a plurality of resource-related data structures, wherein each said resource-related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

coupling an instruction packer to said user preference queue and said resource-related data structures, said instruction packer creates a group of N instructions valid for parallel execution by the golden model and the processor system under test, where N equals 1 or more; and

coupling an instruction generator and simulator to said instruction packer and said resource-related data structures, said instruction generator and simulator generates and simulates instructions that correspond said group of N instructions created by said instruction packer, evaluates the updated architectural state of the golden model, and updates said resource-related data structures.

12. (original). The method of claim 11, wherein said group of N instructions valid for parallel execution further comprises N instructions that do not utilize common system resources other than source registers of the golden model or the processor system under test.

13. (original). The method of claim 11, wherein said instruction packer creates said group of N instructions valid for parallel execution by selecting instructions from an instruction tree in said user preference queue, wherein said instruction packer iteratively creates a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information indicated by said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test.

14. (original). The method of claim 13, wherein said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction.

15. (original). The method of claim 11, wherein said instruction packer creates said group of N instructions valid for parallel execution by selecting instructions in one of the following ways: by selecting each instruction in the order that said instruction appears in an ordered instruction list in said user preference queue, or by selecting a "no operation" instruction where the next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules.

16. (currently amended). A method that generates a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

filling a user preference queue with queue entries, wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

generating a plurality of resource-related data structures, wherein each said resource- related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] <u>two</u> or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

creating a group of N instructions valid for parallel execution by the golden model and the processor system under test, where N equals 1 or more, by selecting instructions from said queue entries based upon information within said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test; and

generating and simulating instructions that correspond said group of N instructions, evaluating the updated architectural state of the golden model, and updating said resource-related data structures.

17. (original). The method of claim 16, wherein said group of N instructions valid for parallel execution further comprises N instructions that do not utilize common system resources other than source registers of the golden model or the processor system under test.

18. (original). The method of claim 16, wherein said group of N instructions valid for parallel execution is created by selecting instructions from an instruction tree in said user preference queue, and said method further comprises iteratively creating a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information within said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test.

19. (original). The method of claim 18, wherein said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction.

20. (original). The method of claim 16, wherein creating said group of N instructions valid for parallel execution further comprises selecting instructions in one of the following ways: selecting each instruction in the order that said instruction appears in an ordered instruction list in said user preference queue, or selecting a "no operation" instruction where the next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules.

21. (currently amended). A program storage device readable by a computer that tangibly embodies a program of instructions executable by the computer to perform a method that generates a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

filling a user preference queue with queue entries, wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

generating a plurality of resource-related data structures, wherein each said resource-related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

creating a group of N instructions valid for parallel execution by the golden model and the processor system under test, where N equals 1 or more, by selecting instructions from said queue entries based upon information within said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test; and

generating and simulating instructions that correspond said group of N instructions, evaluating the updated architectural state of the golden model, and updating said resource-related data structures.

22. (original). The program storage device of claim 21, wherein said group of N instructions valid for parallel execution further comprises N instructions that do not utilize common system resources other than source registers of the golden model or the processor system under test.

23. (original). The program storage device of claim 21, wherein said group of N instructions valid for parallel execution is created by selecting instructions from an instruction tree in said user preference queue, and said method further comprises iteratively creating a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information within said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test.

24. (original). The program storage device of claim 23, wherein said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction.

25. (original). The program storage device of claim 21, wherein creating said group of N instructions valid for parallel execution further comprises selecting instructions in one of the following ways: selecting each instruction in the order that said instruction appears in an ordered instruction list in said user preference queue, or selecting a "no operation" instruction where the next

instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules.

26. (currently amended). A processor verification test apparatus that uses a golden model to generate a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

a user preference queue that comprises queue entries wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

a plurality of resource-related data structures, wherein each said resource-related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

an instruction packer coupled to said user preference queue and said resource-related data structures, said instruction packer creates a group of N instructions valid for parallel execution by the golden model and the processor system under test, wherein said N instructions do not utilize common system resources other than source registers of the golden model or the processor system under test and where N equals 1 or more, by selecting instructions in one of the following two ways: from an instruction tree in said user preference queue, wherein said instruction packer iteratively creates a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information indicated by said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test, in which case said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, and wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction, or from an ordered instruction list in said user preference queue, wherein each instruction selected comprises either the next instruction in said ordered instruction list or a "no operation" instruction if said next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules;

and an instruction generator and simulator that generates and simulates instructions that correspond to said group of N instructions created by said instruction packer, evaluates the updated architectural state of the golden model, and updates said resource-related data structures.

27. (currently amended). A processor verification test system that uses a golden model to generate a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

a user preference queue that comprises queue entries wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

a plurality of resource-related data structures, wherein each said resource-related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

an instruction packer coupled to said user preference queue and said resource-related data structures, said instruction packer creates a group of N instructions valid for parallel execution by the golden model and the processor system under test, wherein said N instructions do not utilize common system resources other than source registers of the golden model or the processor system under test and where N equals 1 or more, by selecting instructions in one of the following two ways: from an instruction tree in said user preference queue, wherein said instruction packer iteratively creates a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information indicated by said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test, in which case said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, and wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction, or from an ordered instruction list in said user preference queue, wherein each instruction selected comprises either the next instruction in said ordered instruction list or a "no operation" instruction if said next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules; and

an instruction generator and simulator that generates and simulates instructions that correspond to said group of N instructions created by said instruction packer, evaluates the updated architectural state of the golden model, and updates said resource-related data structures.

28. (currently amended). A method that makes a processor verification test apparatus that uses a golden model to generate a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

providing a user preference queue that comprises queue entries wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

providing a plurality of resource-related data structures, wherein each said resource- related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

providing an instruction packer coupled to said user preference queue and said resource-related data structures, said instruction packer creates a group of N instructions valid for parallel execution by the golden model and the processor system under test, wherein said N instructions do not utilize common system resources other than source registers of the golden model or the processor system under test and where N equals 1 or more, by selecting instructions in one of the following two ways: from an instruction tree in said user preference queue, wherein said instruction packer iteratively creates a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information indicated by said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test, in which case said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, and wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction, or from an ordered instruction list in said user preference queue, wherein each instruction selected comprises either the next instruction in said ordered instruction list or a "no operation" instruction if said next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules; and

providing an instruction generator and simulator that generates and simulates instructions that correspond to said group of N instructions created by said instruction packer, evaluates the updated architectural state of the golden model, and updates said resource-related data structures.

29. (currently amended). A method that uses a golden model to generate a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

filling a user preference queue that comprises queue entries wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

generating a plurality of resource-related data structures, wherein each said resource- related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

creating a group of N instructions valid for parallel execution by the golden model and the processor system under test using an instruction packer coupled to said user preference queue and said resource-related data structures, wherein said N instructions do not utilize common system resources other than source registers of the golden model or the processor system under test and where N equals 1 or more, by selecting instructions in one of the following two ways: from an instruction tree in said user preference queue, wherein said instruction packer iteratively creates a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information indicated by said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test, in which case said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, and wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction, or from an ordered instruction list in said user preference queue, wherein each instruction selected comprises either the next instruction in said ordered instruction list or a "no operation" instruction if said next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules; and

generating and simulating instructions that correspond to said group of N instructions created by said instruction packer, evaluating the updated architectural state of the golden model, and updating said resource-related data structures.

30. (currently amended). A program storage device readable by a computer that tangibly embodies a program of instructions executable by the computer to perform a method that uses a golden model to generate a test program that verifies that a processor system under test properly executes two or more instructions issued and executed in parallel, comprising:

filling a user preference queue that comprises queue entries wherein each queue entry further comprises an instruction to be tested, a group or tree of instructions to be tested, or a test generator control command;

generating a plurality of resource-related data structures, wherein each said resource-related data structure comprises information concerning selected system resources of the golden model, wherein said information comprises [[~~one~~]] two or more of the following: actual past state, actual present state, actual future state, predicted past state, predicted present state, or predicted future state;

creating a group of N instructions valid for parallel execution by the golden model and the processor system under test using an instruction packer coupled to said user preference queue and said resource-related data structures, wherein said N instructions do not utilize common system resources other than source registers of the golden model or the processor system under test and where N equals 1 or more, by selecting instructions in one of the following two ways: from an instruction tree in said user preference queue, wherein said instruction packer iteratively creates a group of potentially valid instructions by eliminating instructions ineligible for selection, based upon information indicated by said resource-related data structures and the instruction grouping rules for the golden model and the processor system under test, in which case said group of N instructions further comprises at least a first instruction and a second instruction, wherein said second instruction is selected from said group of potentially valid instructions, and wherein said second instruction further comprises an instruction that does not utilize the same system resources other than source registers utilized by said first instruction, or from an ordered instruction list in said user preference queue, wherein each instruction selected comprises either the next instruction in said ordered instruction list or a "no operation" instruction if said next instruction in said ordered instruction list requires unavailable system resources or violates the processor grouping rules; and

generating and simulating instructions that correspond to said group of N instructions created by said instruction packer, evaluating the updated architectural state of the golden model, and updating said resource-related data structures.